FACULDADE DE TECNOLOGIA DE SÃO PAULO

Fernando Guerriero Cardoso da Silva

Levantamento de Requisitos aplicado à Pesquisa e Desenvolvimento de Produtos de Software

SÃO PAULO

2023

FACULDADE DE TECNOLOGIA DE SÃO PAULO

Fernando Guerriero Cardoso da Silva

Levantamento de Requisitos aplicado à Pesquisa e Desenvolvimento de Produtos de Software

Trabalho submetido como exigência parcial
para a obtenção do Grau de Tecnólogo em
Análise e Desenvolvimento de Sistemas
Orientador: Mestre Dionisio Gava Junior

SÃO PAULO

2023

DEDICATÓRIA

Dedico este artigo à minha amada esposa: Juliana Santos da Cruz, quem me ensinou como o amor pode ser, ao mesmo tempo, tranquilo, edificador, excitante e surpreendente!

AGRADECIMENTOS

Primeiramente a minha esposa, Juliana santos da Crus e ao meu orientador, Mestre Dionizio Gava Junior pela ajuda na revisão deste texto.

A Centro Paula Souza e a todos os envolvidos por disponibilizar ensino de qualidade que transformou minha vida.

A meu amigo e ex-supervisor Eric Asis de Oliveira, por me ensinar como exercer de forma admirável liderança, humildade e ética e quem primeiro me ensinou a elicitar requisitos sem nem mesmo conhecer este termo.

A Felipe Esteves Modesto, Nelson Narimatu e Nicholas Shirazawa por me darem a oportunidade de vivenciar a experiência de trabalhar em um setor de Pesquisa e Desenvolvimento de Software.

Lista de Figuras

RESUMO

A produção de um software requer um conjunto de processos que garantam a qualidade do produto final. Dentre esses processos, destaca-se o levantamento de requisitos, que consiste em coletar informações sobre as necessidades e expectativas dos usuários para a criação de um software que atenda suas demandas. Este trabalho tem por objetivo apresentar uma revisão bibliográfica sobre o levantamento de requisitos aplicado à pesquisa e desenvolvimento de produtos de software, destacando as principais técnicas utilizadas nesse processo. Foram consultados livros, artigos científicos e publicações na internet sobre o assunto, com o intuito de compreender as melhores práticas e ferramentas disponíveis para o desenvolvimento de software. Abordaremos o processo de levantamento de requisitos aplicado à pesquisa e desenvolvimento de produtos de software, apresentando um caso prático que ilustra a aplicação desses conceitos. Como resultado, foi possível identificar que o levantamento de requisitos é uma etapa fundamental no processo de desenvolvimento de software, sendo necessário o uso de diversas técnicas para garantir a eficiência e qualidade do produto final.

Palavras-chave: Levantamento de Requisitos; Software; Desenvolvimento.

ABSTRACT

The production of software requires a set of processes that ensure the quality of the final product. Among these processes, the requirements gathering stands out, which consists of collecting information about the needs and expectations of users for the creation of software that meets their demands. This work aims to present a literature review on requirements gathering applied to research and development of software products, highlighting the main techniques used in this process. Books, scientific articles and publications on the Internet were consulted on the subject, in order to understand the best practices and tools available for software development. We will address the process of requirements gathering applied to software product research and development, presenting a practical case that illustrates the application of these concepts. As a result, it was possible to identify that requirements gathering is a fundamental step in the software development process, requiring the use of various techniques to ensure the efficiency and quality of the final product.

Keywords: Requirements Gathering; Software; Development.

Sumário

INTRODUÇÃO

A complexidade do processo de desenvolvimento de software está intrinsecamente ligada à natureza dinâmica e em constante evolução dos requisitos do software, aos diversos atores envolvidos, às restrições tecnológicas e de negócio, bem como aos desafios de comunicação e coordenação entre as partes interessadas. Como ressalta Pressman (2016), “o desenvolvimento de software é um processo intelectualmente desafiador e socialmente complexo, que requer habilidades multidisciplinares, colaboração efetiva e adaptação contínua às mudanças”.

O levantamento de requisitos é o processo de identificar, documentar e verificar as necessidades e expectativas dos usuários para um sistema de software. Envolve a coleta de informações relevantes para entender o contexto, as funcionalidades desejadas e as restrições que devem ser consideradas durante o desenvolvimento do software. Como afirmam Sommerville (2018), “o levantamento de requisitos é a base para a construção de um software que atenda às necessidades dos usuários e seja capaz de agregar valor ao negócio”.

O processo de levantamento de requisitos é essencial para o desenvolvimento de produtos de software de qualidade. Conforme ressaltado por Pfleeger e Atlee (2010), “o sucesso de um projeto de software depende em grande parte da qualidade do levantamento de requisitos, pois é nessa fase que se estabelece a base para todas as atividades subsequentes, como design, implementação e teste”. No entanto, muitas empresas enfrentam desafios para alocar recursos adequados a esse processo. Neste trabalho, apresentamos um caso prático baseado em uma situação real na empresa onde o autor trabalha, que enfrenta problemas de qualidade no produto final. O objetivo é analisar como a falta de recursos dedicados ao processo de levantamento de requisitos pode afetar a qualidade do produto final e apresentar possíveis soluções para esse problema.

1. PESQUISA E DESENVOLVIMENTO DE PRODUTOS DE SOFTWARE

A pesquisa e desenvolvimento de produtos de software envolve o processo de investigação, criação, aprimoramento e implementação de novos produtos ou funcionalidades de software. Segundo Sommerville (2011, p. 222), "a pesquisa e desenvolvimento de produtos de software compreende a atividade de buscar, criar e implementar novos produtos de software, ou melhorar os produtos existentes, para atender às necessidades dos usuários e às demandas do mercado".

A pesquisa e desenvolvimento de produtos de software é um processo essencial para a inovação e evolução dos produtos de software. Beath, Becerra-Fernandez e Ross (2013) afirmam que "a pesquisa e desenvolvimento de produtos de software são atividades fundamentais para as organizações que buscam se manter competitivas no mercado de tecnologia, permitindo a criação de novos produtos, aprimoramento dos produtos existentes e a identificação de oportunidades de mercado".

A pesquisa e desenvolvimento de produtos de software normalmente envolve diferentes etapas e atividades, como a identificação de necessidades dos usuários, a definição de requisitos, o design, a implementação, os testes e a implantação de novos produtos ou funcionalidades de software. É um processo iterativo e colaborativo, que requer a participação de equipes multidisciplinares e a aplicação de métodos e técnicas adequadas para garantir a qualidade e a eficácia do produto de software resultante. Pressman (2016, p. 70) descreve o seguinte conjunto de etapas: identificação de necessidades, definição de requisitos, design, implementação, testes, implantação, manutenção. A seguir conceituaremos e contextualizaremos estas etapas.

1.1. Identificação de necessidades

Pressman (2016, p. 70) descreve a identificação de necessidades como uma etapa crucial do processo de engenharia de software, destacando que "a identificação de necessidades é o processo de entender os problemas do usuário, oportunidades de negócio ou demandas do mercado que podem ser atendidos pela aplicação de um novo produto de software". Ele também enfatiza que essa etapa envolve a coleta de informações dos stakeholders relevantes, como usuários finais, clientes, patrocinadores, entre outros, a fim de compreender suas necessidades e expectativas em relação ao produto de software em questão. Essa etapa é fundamental no processo de pesquisa e desenvolvimento de produtos de software, pois permite compreender as necessidades reais dos usuários e do mercado, orientando as decisões subsequentes do projeto de software.

Esta primeira etapa também deve envolver a compreensão do contexto de mercado, análise da concorrência, identificação de lacunas existentes em produtos de software existentes e a compreensão de tendências e demandas do mercado atual e futuro. É uma etapa crítica para garantir que o produto de software a ser desenvolvido atenda às reais necessidades dos usuários e às demandas do mercado, visando a satisfação dos clientes e o sucesso do produto no mercado.

A figura responsável por esta etapa do desenvolvimento de produtos de software pode variar de acordo com a organização e o contexto do projeto. No entanto, em muitos casos, é comum que um analista de negócios desempenhe esse papel. O analista de negócios é responsável por compreender as necessidades dos usuários, realizar entrevistas, análise de documentos e interações com as partes interessadas para identificar, documentar e aprovar os requisitos do software.

1.2. Definição de requisitos

A definição de requisitos é uma etapa essencial no processo de pesquisa e desenvolvimento de produtos de software, onde as necessidades identificadas são especificadas em requisitos claros, precisos e completos, que servirão de base para o projeto e implementação do software. A definição de requisitos envolve a documentação e especificação detalhada das funcionalidades, características, comportamentos, desempenho e outras características relevantes do software.

Sommerville (2018, p. 139) descreve a definição de requisitos como "o processo de estabelecer os serviços que o cliente exige de um sistema e as restrições sob as quais ele opera e é desenvolvido". Ele enfatiza que a definição de requisitos é uma etapa crítica para garantir que todos os aspectos relevantes do software sejam claramente definidos e compreendidos por todas as partes interessadas, incluindo os desenvolvedores de software, clientes, usuários finais e demais stakeholders.

Esta etapa deve incluir a identificação e especificação de requisitos funcionais, que se referem às funcionalidades e comportamentos esperados do software, e requisitos não funcionais, que se referem a características como desempenho, segurança, usabilidade, confiabilidade, entre outros. Além disso, a definição de requisitos deve envolver a validação dos requisitos com os stakeholders relevantes, para garantir que todas as expectativas sejam atendidas e que o software seja desenvolvido de acordo com as necessidades e demandas identificadas.

A definição de requisitos é uma atividade colaborativa e geralmente envolve a participação de várias partes interessadas. Não há uma figura específica responsável exclusivamente pela definição de requisitos, pois o processo geralmente requer a contribuição de analistas de negócios, desenvolvedores, gerentes de projeto, usuários finais e outras partes envolvidas no projeto de software.

> *"Os requisitos são definidos pelos usuários, clientes e outros stakeholders, com a ajuda de analistas de requisitos e engenheiros de software. A definição dos requisitos é uma atividade colaborativa que envolve a comunicação e negociação entre todos os envolvidos no projeto de software."*
>
> *(PRESSMAN, 2016)*

Durante a definição dos requisitos, deve-se também, definer quais serão os critérios de aceite dos requisitos, como serão feitos os testes e quais ferramentas serão necessárias para realizar esta validação (infraestrutura, acessos, etc). Estes detalhes são importantes para evitar atrasos de entregas gerados pela falta destas informações. Estes acordos devem sempre ser formalizados e devidamente documentados para resguardar a responsabilidade do levantador de requisitos sobre recursos que ele mesmo não tem poder ou conhecimento.

1.3. Design

O design, no contexto da pesquisa e desenvolvimento de produtos de software, refere-se à criação de soluções de software que atendam aos requisitos definidos, considerando aspectos como a arquitetura do software, a estrutura de dados, a interface do usuário, a usabilidade, a experiência do usuário e outros elementos relevantes para o desenvolvimento do software.

Pressman (2016, p. 239) define o design de software como "o processo de definir a arquitetura, componentes, módulos, interfaces e dados para um sistema para satisfazer os requisitos especificados". Ele ressalta que o design é uma etapa crítica do desenvolvimento de software, onde a solução técnica é elaborada com base nos requisitos definidos, e é uma atividade que deve envolver a criação de modelos, diagramas, protótipos e outras representações visuais do software em desenvolvimento.

O design também deve incluir a definição de padrões de codificação, estratégias de testes, seleção de tecnologias e ferramentas de desenvolvimento, bem como a consideração de fatores como a manutenibilidade, escalabilidade, robustez e outros atributos de qualidade do software. O objetivo do design é criar uma solução de software que seja eficiente, eficaz, confiável, fácil de usar e que atenda às necessidades dos usuários e às expectativas dos stakeholders.

A figura responsável pelo "Design" no contexto de desenvolvimento de software pode variar dependendo da abordagem e estrutura organizacional de cada projeto. Em equipes maiores, é comum haver papéis específicos, como designer de interface, designer de interação ou arquiteto de software, que desempenham um papel fundamental na fase de design.

> *"A criação de um design eficaz requer a colaboração e o envolvimento de várias partes interessadas, como designers, desenvolvedores, especialistas no domínio do problema e usuários finais. Essas partes interessadas devem trabalhar juntas para criar soluções que atendam aos requisitos e necessidades do sistema."*
> *(SOMMERVILLE, 2011)*

1.4. Implementação e/ou Construção

A implementação é a fase do processo de desenvolvimento de software em que o código-fonte é escrito e as soluções de software são construídas de acordo com o design previamente elaborado. Nesta fase, as instruções de programação são traduzidas em código-fonte por meio de linguagens de programação, ferramentas e ambientes de desenvolvimento.

De acordo com Sommerville (2018, p. 252), a implementação é a "construção do sistema de software", onde o código-fonte é criado com base nos requisitos e no design, definidos anteriormente. É uma etapa crucial do desenvolvimento de software, onde os programadores trabalham na codificação, testes unitários e integração dos componentes do software para criar a versão executável do produto.

A implementação deve envolver a utilização de metodologias de desenvolvimento, como o desenvolvimento iterativo e incremental, programação em pares, revisões de código, dentre outras práticas, para garantir a qualidade e a eficiência do código produzido. Além disso, deve-se utilizar de ferramentas de controle de versão, depuração, refatoração e outras técnicas para garantir a integridade e a manutenibilidade do código-fonte.

A figura responsável pela "Implementação" no contexto de desenvolvimento de software é geralmente desempenhada pelos programadores ou desenvolvedores de software. São eles que traduzem o design e os requisitos do software em código executável.

"Os programadores são responsáveis por implementar os requisitos do software, traduzindo o design em código executável. Eles devem possuir habilidades técnicas sólidas e conhecimento das melhores práticas de programação para garantir a qualidade e eficiência do software resultante." (PRESSMAN, 2016)

1.5. Testes

Os testes são atividades realizadas para verificar e validar se o software está funcionando de acordo com os requisitos definidos, identificar defeitos e garantir a qualidade do produto final. Os testes devem ser realizados em diferentes níveis, como testes unitários (que podem ser feitos pelos próprios programadores durante a fase de implementação), testes de integração, testes de aceitação (feitos pelo próprio cliente), entre outros, e devem envolver diferentes técnicas, como testes funcionais e não funcionais, testes de desempenho, testes de segurança, entre outros. Importante salientar que os testes de desenpenho e de segurança devem ser realizados por especialistas no assunto para que a validação sea feita de forma satisfatória já que se tratam de assuntos delicados e muito especificos.

De acordo com Pressman (2016, p. 595), os testes são "atividades de verificação de software que têm como objetivo descobrir falhas no software", visando garantir a qualidade do software produzido. Os testes são planejados e executados em diferentes fases do ciclo de vida do software, e são fundamentais para identificar e corrigir defeitos antes de o software ser disponibilizado aos usuários finais.

Os testes podem ser automatizados ou manuais, e podem envolver a criação de casos de teste, a execução dos casos de teste, a análise dos resultados e a correção dos defeitos identificados. Além disso, os testes também são utilizados como parte do processo de validação do software, garantindo que o produto atenda às necessidades dos usuários e aos requisitos definidos.

A figura responsável pelos "Testes" no contexto de desenvolvimento de software é o profissional de qualidade de software ou tester. Eles são responsáveis por garantir que o software atenda aos requisitos e funcionalidades esperados, identificar e reportar erros e problemas, além de verificar a qualidade geral do sistema.

"Os profissionais de qualidade de software têm a tarefa de realizar testes para verificar se o software atende aos requisitos estabelecidos, garantindo que o produto final seja confiável, funcional e livre de erros." (SOMMERVILLE, 2011)

1.6. Implantação

A implantação refere-se à fase em que o software é instalado, configurado e disponibilizado para uso em um ambiente de produção. É o processo de tornar o software pronto para ser utilizado pelos usuários finais, com todos os componentes e recursos necessários para seu pleno funcionamento.

De acordo com Pressman (2016, p. 625), a implantação é "o processo de instalação do software, incluindo todos os elementos necessários para que ele possa ser utilizado de forma efetiva no ambiente de operação do usuário final". Essa fase envolve a configuração do software em servidores, a instalação de bancos de dados, a realização de testes finais, a migração de dados, a configuração de ambientes de produção, entre outras atividades.

A implantação também envolve a disponibilização do software aos usuários finais, a realização de treinamentos (caso a funcionalidade a ser implantada seja muito complexa e/ou extensa, os treinamentos devem acontecer com certa antecedência à implantação, caso contrário os usuários poderão não estar prontos quando a funcionalidade estiver disponível em produção), a configuração de permissões de acesso, a configuração de interfaces com outros sistemas, entre outras tarefas necessárias para que o software esteja plenamente funcional em um ambiente de produção.

A figura responsável pela "Implantação" no contexto de desenvolvimento de software, geralmente é desempenhado por profissionais de implantação e/ou infraestrutura, engenheiros de sistemas ou administradores de sistemas. Eles são responsáveis por garantir que o software seja instalado, configurado e esteja funcionando corretamente nos ambientes de produção.

1.7. Manutenção

A manutenção, no contexto da pesquisa e desenvolvimento de produtos de software, refere-se às atividades realizadas após a implantação do software com o objetivo de corrigir defeitos, realizar melhorias, adaptar o software a novos requisitos ou corrigir problemas de desempenho.

De acordo com Sommerville (2011, p. 633), a manutenção é "o processo de modificar um sistema depois que ele foi colocado em uso". Essa fase envolve a identificação e correção de defeitos (ou "bugs") no software, a realização de atualizações e melhorias para acompanhar as mudanças nas necessidades dos usuários ou nas tecnologias, e a correção de problemas de desempenho, entre outras atividades.

A manutenção é classificada em diferentes tipos, como manutenção corretiva (correção de defeitos), manutenção adaptativa (adaptação a novos requisitos), manutenção evolutiva (melhorias funcionais) e manutenção preventiva (ações para evitar futuros problemas). A manutenção é uma fase essencial do ciclo de vida de um software, uma vez que o software, tal como o negócio, estará em constante evolução e necessitará de ajustes e atualizações ao longo do tempo.

A figura responsável pela "Manutenção" no desenvolvimento de software serão atribuída aos profissionais de manutenção de software, engenheiros de manutenção, lideres técnicos e desenvolvedoresde software. Esses profissionais são responsáveis por realizar atividades de correção de bugs, atualizações, melhorias e suporte contínuo ao software após o seu lançamento.

2. LEVANTAMENTO DE REQUISITOS

2.1. Conceito

"Levantamento de Requisitos" pode ser conceituado como um processo no desenvolvimento de software que envolve a identificação, análise e documentação das necessidades, expectativas e restrições dos stakeholders (envolvidos) de um sistema de software, visando compreender e especificar claramente o que o sistema deve ser capaz de fazer. Segundo Pressman (2016, p. 117), "o levantamento de requisitos é uma atividade essencial no processo de desenvolvimento de software, pois representa a base para a definição do escopo do projeto e o estabelecimento de um entendimento comum entre os stakeholders sobre o que deve ser construído". Nesse sentido, o levantamento de requisitos é uma etapa crítica para o sucesso de projetos de desenvolvimento de software, pois estabelece as bases para o desenvolvimento de um sistema que atenda às expectativas e necessidades dos usuários e demais stakeholders envolvidos.

O levantamento de requisitos é um processo que consiste em coletar informações sobre as necessidades e expectativas dos usuários em relação a um software que será desenvolvido. Esse processo é fundamental para o sucesso do projeto, pois define todas as funcionalidades e características que o software deve possuir.

Este processo deve ser realizado de forma sistemática e organizada, com o objetivo de identificar todas as necessidades dos usuários e transformá-las em um conjunto claro e preciso de requisitos a serem cumpridos pelo

2.2. Objetivos

De acordo com Sommerville (2018, p. 85), "os objetivos do levantamento de requisitos são entender os problemas do cliente e a oportunidade de negócio, descobrir e compreender os requisitos do sistema e estabelecer as bases para o contrato do sistema". Pressman (2016, p. 117) também enfatiza que o objetivo principal do levantamento de requisitos é "definir o que o sistema de software deverá fazer e estabelecer um acordo mútuo sobre os recursos, desempenho e outras características do sistema". Além disso, alguns outros objetivos do levantamento de requisitos incluem:

- Identificar as necessidades e expectativas dos stakeholders do sistema;
- Compreender e documentar os requisitos funcionais e não funcionais do sistema;
- Definir os limites e o escopo do sistema a ser desenvolvido;
- Estabelecer uma base sólida para o desenvolvimento de um sistema que atenda às expectativas dos usuários e stakeholders;
- Minimizar a ocorrência de erros e mudanças de escopo durante o desenvolvimento do sistema, por meio de uma compreensão clara dos requisitos.

2.3. Técnicas de Levantamento de Requisitos

Barbosa e Silva (2010, p. 67) dizem que "as técnicas de levantamento de requisitos são métodos, práticas ou abordagens utilizadas para coletar, analisar e documentar os requisitos de um sistema de software". Pressman (2016, p. 118) também destaca que "as técnicas de levantamento de requisitos incluem entrevistas, questionários, observação de usuários, análise de documentos, prototipagem, entre outros métodos, que visam obter informações detalhadas sobre o sistema a ser desenvolvido".

Abaixo abordaremos algumas das principais técnicas de levantamento de requisites.

2.3.1. Entrevistas

Esta técnica é amplamente utilizada na Engenharia de Requisitos para coletar informações diretamente dos stakeholders do sistema a ser desenvolvido. Segundo Pressman (2016, p. 119), "a entrevista é uma técnica de levantamento de requisitos que envolve a obtenção de informações por meio de questionamentos diretos a um ou mais stakeholders do sistema, com o objetivo de identificar suas necessidades, expectativas e restrições em relação ao sistema".

Outros autores também destacam a importância das entrevistas como uma técnica fundamental de levantamento de requisitos. Por exemplo, Sommerville (2018) ressalta que "a entrevista é uma técnica de levantamento de requisitos amplamente utilizada, pois permite aos analistas obterem informações detalhadas diretamente dos stakeholders, possibilitando a compreensão das necessidades e expectativas dos usuários e demais envolvidos no sistema".

As entrevistas podem ser conduzidas de forma estruturada, com um roteiro de perguntas pré-definido, ou de forma não estruturada, permitindo uma abordagem mais flexível e aberta às respostas dos stakeholders. É importante considerar a seleção dos participantes da entrevista, a definição dos objetivos da entrevista, a elaboração de perguntas relevantes e a documentação adequada dos resultados obtidos.

2.3.2. Questionários

A técnica de levantamento de requisitos denominada "Questionários" é uma abordagem na qual os stakeholders do sistema respondem a perguntas previamente elaboradas, com o objetivo de coletar informações sobre suas necessidades, expectativas e restrições em relação ao sistema em desenvolvimento. Segundo Leite et al. (2017, p. 85), "o questionário é uma técnica de levantamento de requisitos que permite obter informações de forma padronizada e geralmente escrita, por meio de perguntas formuladas aos stakeholders".

Outros autores também ressaltam a utilidade dos questionários como uma técnica eficaz para coletar dados quantitativos e qualitativos dos stakeholders. Por exemplo, Pohl et al. (2011) destacam que "os questionários são uma técnica amplamente utilizada em levantamento de requisitos, pois permitem coletar informações de uma grande quantidade de stakeholders de forma sistemática e padronizada".

Os questionários podem ser elaborados de forma fechada, com respostas pré-definidas, ou aberta, permitindo que os stakeholders expressem suas opiniões de forma livre. É importante considerar o desenho do questionário, a seleção dos participantes, a análise dos resultados obtidos e a garantia da confidencialidade das respostas dos stakeholders.

2.3.3. Observação

A técnica de levantamento de requisitos denominada "Observação" consiste em observar diretamente o ambiente de trabalho dos stakeholders, suas atividades e interações, com o objetivo de identificar requisitos implícitos ou não documentados. Segundo Sommerville (2011, p. 107), "a observação é uma técnica de levantamento de requisitos que envolve a observação direta do comportamento de usuários ou do ambiente em que o sistema será utilizado".

Outros autores também destacam a importância da observação como uma técnica valiosa para identificar requisitos que podem não ser capturados por meio de entrevistas ou questionários. Por exemplo, Robertson e Robertson (2012) afirmam que "a observação é uma técnica poderosa para identificar requisitos implícitos e não documentados, pois permite aos analistas verem como os usuários reais usam o sistema em seu ambiente de trabalho".

A observação deve ser realizada de forma participativa, na qual o analista de requisitos interage com os stakeholders durante suas atividades, ou de forma não participativa, apenas observando sem interagir diretamente. É importante considerar a ética e a privacidade dos stakeholders durante a observação, garantindo que suas atividades sejam observadas de forma não intrusiva e respeitando a confidencialidade das informações obtidas.

2.3.4. Análise de documentos

Esta técnica consiste em analisar documentos existentes, tais como documentos de requisitos, especificações técnicas, manuais, relatórios, entre outros, para identificar e extrair os requisitos relevantes do sistema. De acordo com Pressman (2016, p. 196), "a análise de documentos é uma técnica de levantamento de requisitos que envolve a revisão e análise de documentos escritos, tais como documentos de requisitos, especificações técnicas, manuais de usuários, entre outros, para identificar requisitos".

A análise de documentos destaca-se como uma técnica eficaz para identificar requisitos, especialmente quando se trata de sistemas legados ou sistemas que já estão em operação. Wiegers e Beatty (2013) afirmam que "a análise de documentos é uma técnica valiosa para extrair requisitos de sistemas legados ou sistemas em operação, pois muitas vezes os requisitos já estão documentados em manuais, especificações técnicas ou outros documentos".

Durante a análise de documentos, o analista de requisitos examina cuidadosamente o conteúdo dos documentos, identificando informações relevantes, como requisitos funcionais, requisitos não funcionais, restrições, interfaces, entre outros, e os registra em uma lista consolidada de requisitos. É importante verificar a confiabilidade e atualidade dos documentos analisados, garantindo que as informações obtidas sejam precisas e atualizadas.

2.3.5. Prototipação

A prototipação é uma técnica utilizada no processo de desenvolvimento de software que tem como objetivo criar uma versão preliminar do produto para ser testada e avaliada pelos usuários finais, a fim de validar os requisitos levantados e verificar se as soluções propostas atendem às suas necessidades. De acordo com Pressman (2016, p. 187), prototipação é "um processo iterativo que constrói um modelo em escala reduzida de uma aplicação para descobrir e validar requisitos".

A prototipação é uma técnica importante no levantamento de requisitos, pois permite que os usuários finais visualizem e interajam com a aplicação antes mesmo de sua implementação, possibilitando a identificação de possíveis problemas ou necessidades que não foram considerados durante o processo de elicitação de requisitos (Sommerville, 2011).

Além disso, a prototipação também ajuda a reduzir o tempo e os custos do desenvolvimento, já que permite que os requisitos sejam refinados e validados antes da implementação completa do software.

Em resumo, a prototipação é uma técnica importante no processo de levantamento de requisitos, permitindo a validação e refinamento dos requisitos antes da implementação completa do software, reduzindo o tempo e os custos do desenvolvimento e aumentando a satisfação dos usuários finais.

3. DESAFIOS NA ELICITAÇÃO DE REQUISITOS: um Caso Prático Na Empresa Hardware Manufacturing Ltda.

A seguir será apresentado um estudo de caso de um setor de Pesquisa e Desenvolvimento. Neste capítulo, será realizada uma contextualização da empresa, identificação dos problemas enfrentados na identificação de requisitos, análise dos motivos desses problemas, proposta de solução para aprimorar o processo de levantamento de requisitos, implementação da solução e a conclusão do caso prático. Através da análise desse caso, é possível compreender os desafios enfrentados na identificação de requisitos e como a implementação de soluções pode ajudar a melhorar esse processo.

Ao longo deste capítulo, utilizaremos o nome fictício “Hardware Manufacturing Ltda.” para se referir à empresa onde o caso prático foi observado. Essa medida foi tomada para preservar a confidencialidade da organização real e garantir a privacidade dos envolvidos.

3.1. Contextualização da empresa

A Hardware Manufacturing Ltda é uma multinacional com expertise no ramo da manufatura de hardware. Originária de Taiwan, seu corpo executivo e clientes são predominantemente taiwaneses. A empresa possui uma estrutura hierárquica tradicional, mas recentemente, criou o setor de Pesquisa e Desenvolvimento com o objetivo de criar produtos tecnológicos de hardware e software para fortalecer sua presença no mercado brasileiro.

O setor de Pesquisa e Desenvolvimento da Hardware Manufacturing Ltda é composto por um time de 8 pessoas com conhecimentos multidisciplináres em áreas como hardware, eletrônica, engenharia mecânica e engenharia de produção, firmware, desenvolvimento de software, programação, arquitetura de software e gestão de projetos. O setor é relativamente novo, e a equipe tem trabalhado duro para desenvolver produtos inovadores que atendam às necessidades dos clientes brasileiros e fortaleçam a marca da empresa no mercado.

3.2. Identificação dos problemas do caso de estudo

O caso deste estudo, foca em uma funcionalidade de um sistema em que tem como objetivo agilizar a triagem de pacientes e auxiliar o trabalho dos médicos durante seus atendimentos, bem como acompanhar sinais vitais como pressão arterial, oxigenação, temperatura, peso e altura. A ideia deste sistema surgiu a partir de um médico que trabalha no ambulatório médico da empresa em estudo. O projeto deste produto já está em seu quarto ano, sendo que nos dois primeiros anos foi criada a primeira versão de um totem de autoatendimento que coleta os sinais vitais de um paciente. Esta primeira versão utilizava periféricos com tecnologia Bluetooth e foi criada a primeira versão de um aplicativo móvel e uma plataforma gerencial.

Em seu terceiro ano, foi criada uma nova versão do totem de autoatendimento que utiliza periféricos USB, conseguindo melhor performance da máquina e diminuindo o tempo de uso do equipamento. Nessa fase, o aplicativo móvel, a plataforma gerencial e a interface gráfica do totem de autoatendimento foram reformuladas. No entanto, para a segunda versão do totem e dos softwares, a produção, testes e projetos foram terceirizados para institutos de pesquisa externos à Hardware Manufacturing Ltda. Essa terceirização foi feita com o objetivo de aproveitar a expertise desses institutos e garantir a qualidade final do produto.

O levantamento de requisitos é fundamental para o sucesso de qualquer projeto de desenvolvimento de software, especialmente no caso deste projeto de sistema médico. Isso porque a identificação correta das necessidades e expectativas dos usuários finais, médicos e pacientes, é crucial para garantir que o produto desenvolvido atenda às suas necessidades e seja eficaz em sua função.

Além disso, no caso específico deste projeto, o fato de terceirizar a produção, testes e projetos para institutos de pesquisa externos torna ainda mais crucial o processo de levantamento de requisitos, pois é preciso garantir que todas as partes envolvidas tenham uma compreensão clara e precisa do que se espera do produto e como ele deve ser desenvolvido.

Durante a fase de implementação da segunda versão do totem de autoatendimento e dos softwares, a equipe de desenvolvimento da Hardware Manufacturing Ltda identificou alguns problemas no processo de levantamento de requisitos. A funcionalidade de "Configuração da classificação de risco", que permite a configuração dos critérios de classificação de risco dos pacientes, foi uma das principais funcionalidades que apresentaram problemas durante o processo de desenvolvimento.

Durante os testes e validações da funcionalidade em questão, foi identificado que na primeira versão do software, a funcionalidade não atendia às necessidades reais dos médicos e enfermeiras do ambulatório. Portanto, era necessário identificar os problemas para melhorá-la na segunda versão.

Durante a fase de identificação de problemas, foram realizadas entrevistas com os usuários do software, como médicos e enfermeiras, a fim de entender suas necessidades e expectativas em relação à funcionalidade de classificação de risco. Também foram avaliados os fluxos de trabalho dos usuários e como eles interagiam com o software em relação a essa funcionalidade.

Após essas análises, foram identificados alguns problemas na funcionalidade, tais como:

- Dificuldades na identificação das prioridades dos pacientes;
- Critérios para classificação de risco não atendiam as regras do negócio;
- Interface pouco intuitiva para a configuração da classificação de risco.

Abaixo na **Figura 1** podemos ver a tela de configuração do protocolo de risco implementada conforme levantamento de requisitos realizada na primeira fase do projeto. Ela contém 5 abas, uma para cada classe de risco (Emergência, Muito Urgente, Urgente, Pouco urgente e Não urgente), e em cada aba 21 caixas de texto para preenchimento dos valores desta classificação para cada sinal vital: pressão diastólica, pressão sistólica, batimentos cardíacos, temperatura, oxigenação, frequência respiratória, altura, peso, IMC e idade. Cada sinal vital tem duas caixas de texto (mínimo e máximo) exceto para o batimento cardíaco, que contêm quatro caixas, e a idade que contém apenas uma (a idade apenas classifica com maior ou igual a um valor), e o batimento cardíacoo permitia dois intervalos para a classificação, enquanto todos os outros sinais vitais apenas um intervalo.

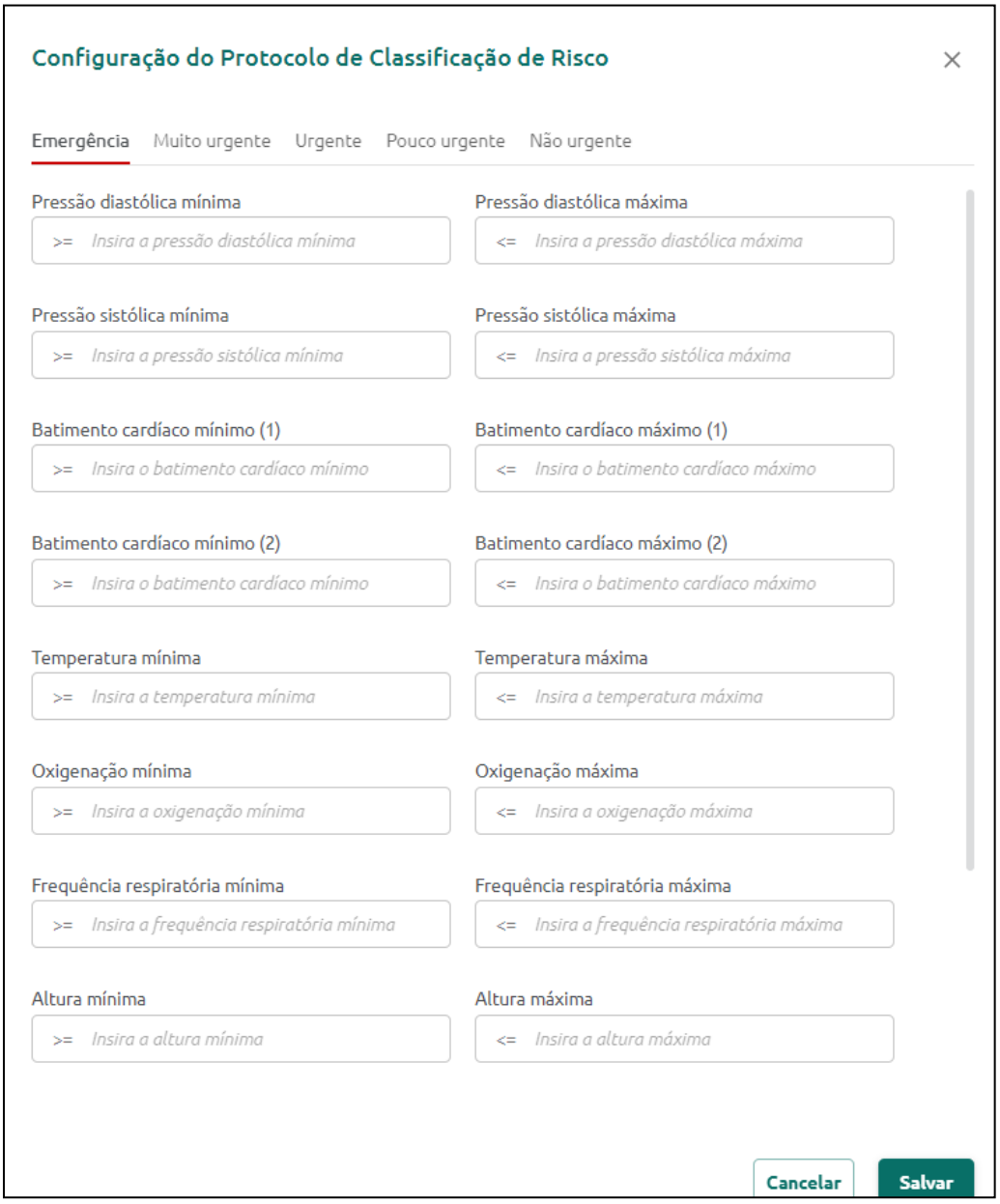

Figura 1: Tela de configuração do protocólo de risco primeira versão

Esses problemas impactaram diretamente no trabalho dos usuários e na qualidade do atendimento aos pacientes, esta funcionalidade ficou desativada por alguns meses, aumentando o trabalho de análise dos sinais vitais do ambulatório. Portanto, era necessário encontrar uma solução para melhorar essa funcionalidade.

No próximo tópico, será discutido a análise dos motivos desses problemas e como foram propostas soluções para melhorar a funcionalidade de "Configuração da classificação de risco" na segunda versão do software.

3.3. Análise dos motivos dos problemas

Após identificar os problemas na funcionalidade de "Configuração da classificação de risco" durante o desenvolvimento da segunda versão do software, foi necessário analisar os motivos que levaram a esses problemas.

Um dos principais motivos foi a falta de um levantamento de requisitos adequado na primeira versão do software. Na época, a funcionalidade foi implementada com base em algumas solicitações informais de um médico e alguns executivos, sem um estudo mais aprofundado das necessidades reais dos usuários (médicos e enfermeiras). Nesta época a figura de um analista de negócio não existia no time de projetos, dessa forma a responsabilidade das tarefas de elicitação de requisitos não eram delegadas e cairam no esquecimento.

Com a reformulação do projeto, algumas pessoas que haviam participado da primeira fase do desenvolvimento do software já não estavam mais na empresa. Ademais, um dos problemas identificados foi a falta da figura de um levantador de requisitos na equipe interna de desenvolvimento de software no setor de Pesquisa e Desenvolvimento, o que levou à alocação desta tarefa a colaboradores dos projetos. Porém, muitas vezes esses colaboradores não possuíam um conhecimento adequado para elicitar requisitos e atender as necessidades dos usuários finais, o que acabou contribuindo para a falta de alinhamento entre as funcionalidades desenvolvidas e as necessidades reais dos usuários.

O principal método utilizado para elicitar requisitos na primeira versão do software era o brainstorm, que aconteciam semanalmente durante reuniões longas com vários colaboradores do projeto e de várias áreas de conhecimento, todos eram livres para opinar e apresentar ideias. A ideia deste formato era boa, mas a execução sem um mediador alongava o tempo da reunião e as ideias não eram registradas e organizadas.

Outro fator relevante que pode ter influenciado ao resultado final da funcionalidade "Configuração da classificação de risco", foi a falta de um processo estruturado de testes e validação da primeira versão da funcionalidade. A figura do tester não havia sido criada no time, e isso pode ter levado a problemas não identificados durante o processo de desenvolvimento e que foram encontrados apenas depois da implantação.

Com base nessa análise dos motivos dos problemas, foi possível identificar os pontos críticos e pensar em soluções para evitar que esses problemas acontecessem novamente.

O próximo passo foi propor uma solução para aprimorar a funcionalidade de "Configuração da classificação de risco".

3.4. Proposta de solução

O objetivo deste subcapítulo é apresentar a proposta de solução para aprimorar a funcionalidade de "Configuração do protocolo de risco". A solução foi desenvolvida a partir de quatro fatores principais:

- Reformulação da interface gráfica da tela de "Configuração do protocolo de risco": Após a identificação dos problemas na funcionalidade na primeira versão do software, foi identificado que a interface gráfica da tela de configuração do protocolo de risco era confusa e pouco intuitiva. Por isso, propôs-se uma reformulação da interface gráfica, visando torná-la mais amigável e fácil de usar.
- Levantamento de requisitos: Foi realizado um novo levantamento de requisitos para a funcionalidade, a partir de entrevistas com os usuários da funcionalidade (médicos e enfermeiras). Essas entrevistas permitiram entender melhor as necessidades e expectativas dos usuários em relação à funcionalidade, possibilitando uma solução mais adequada e alinhada com as necessidades reais dos usuários.
- Redesenho do modelo de configuração: Durante uma das entrevistas, uma enfermeira e um médico sugeriram um novo modelo de configuração, que levou em conta as necessidades e expectativas dos usuários. Esse novo modelo foi então utilizado para desenvolver um protótipo no Figma, que foi submetido à aprovação dos usuários.
- Implantação de um processo de testes mais estruturado: Para a segunda versão do software, foi implantado um processo de testes mais estruturado, que ajudou a validar a nova proposta. Esse processo permitiu identificar e corrigir problemas e erros no desenvolvimento da funcionalidade, antes da sua implantação no ambiente de produção.

Com base nesses fatores, foi proposta uma solução para aprimorar a funcionalidade de "Configuração do protocolo de risco". A nova solução contou com uma interface gráfica mais intuitiva e amigável, um modelo de configuração que atendeu às necessidades dos usuários e foi validado por eles, além de um processo de testes mais estruturado, que permitiu identificar e corrigir problemas antes da implantação.

Inicialmente utilizamos a primeira versão da interface da Configuração da Classificação de risco como base para o novo levantamento, e fizemos uma entrevista inicial com uma enfermeira para entender como a triagem funcionava e quais os quesitos que eram seguidos para a a classificação. A **Figura 2** é o resultado do estudo inicial, feita em uma planilha Excel.

	Emergência	Não urgente		Pouco Urgente		Urgente		Muito Urgênte		Emergência	NÃO CLASSIFICADO
	<=	Min	Max	Min	Max	Min	Max	Min	Max	>=	
PA Sistólica											Valores que não se encaixarem em nenhuma classificação ascima, serão categorizados como "Não Classificado", receben cor cinza.
PA Diastólica											
Bat. Card.											
Temperatura											
Oxigenação											
FR											
Altura											
Peso											
IMC											
Idade											

Figura 2: Primeira Versão do Modelo da Configuração da Classificação de Risco

Em seguida foram feitas entrevistas com mais dois médicos do ambulatório da Hardware Manufacturer Ltda, e foram feitas algumas pesquisas para melhor entendimento do funcionamento dos métodos de mensuração de sinais vitais. Ao fim desta etapa, identificamos que todos os sinais vitais (exceto oxigenação e idade) deveriam ter dois intervalos de valores para cada classificação (limites inferior e superior de cada classe) conforme **Figura 3** abaixo.

	Emergência	Muito Urgênte		Urgente		Pouco Urgente		Não urgente		Pouco Urgente		Urgente		Muito Urgênte		Emergência	Não Classificado
	<=	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	>=	
PA Sistólica																	Aferições de sinal vital com valores que não se encaixam em nenhuma classificação, serão classificados como "Não Classificado"
PA Diastólica																	
Bat. Card.																	
Temperatura																	
Oxigenação																	
FR																	
Altura																	
Peso																	
IMC																	
Idade																	

Figura 3: Versão final do Modelo da Configuração da Classificação de Risco

Após a aprovação dos dois médicos que participaram das entrevistas sobre o modelo de Configuração da Classificação de risco conforme a **Figura 3**, o modelo foi enviado para a designer do instituto responsável pelo desenvolvimento deste software, e ela criou uma proposta de design e apresentou o protótipo abaixo (**Figura 4)**.

Figura 4: Protótipo Figma Seguindo Novo Modelo

O Protótipo também foi submetido a aprovação do Gerente do projeto, pelo desenvolvedor, pelo líder técnico e pelo gerente de software internos da equipe de Pesquisa e Desenvolvimento e também pelos Médicos do ambulatório.

3.5. Implementação da solução

Para a implementação da solução proposta para a funcionalidade de "Configuração da classificação de risco", foram realizadas diversas etapas.

Inicialmente, foi feita uma reformulação da interface gráfica da tela de "configuração do protocolo de risco", buscando torná-la mais intuitiva e de fácil compreensão pelos usuários. Essa reformulação foi baseada em entrevistas com os usuários da funcionalidade e teve como objetivo atender às suas necessidades e expectativas em relação à configuração do protocolo de risco.

A partir dessas entrevistas, foi feito um novo levantamento de requisitos, visando aprimorar a funcionalidade. Durante uma dessas entrevistas, com a participação de uma enfermeira e um médico, foi redesenhado o modelo de configuração, visando a torná-lo mais simples e prático de usar. Após essa etapa, foi desenvolvido um protótipo no Figma, que foi submetido à aprovação dos usuários. A partir dos feedbacks recebidos, o protótipo foi refinado e aprovado.

Com o modelo de configuração definido, foi iniciada a etapa de desenvolvimento da funcionalidade. Para garantir a qualidade do software, foi implantado um processo estruturado de testes, que ajudou a validar a nova proposta. Abaixo na **Figura 5** temos o print do software final sendo executado em um browser.

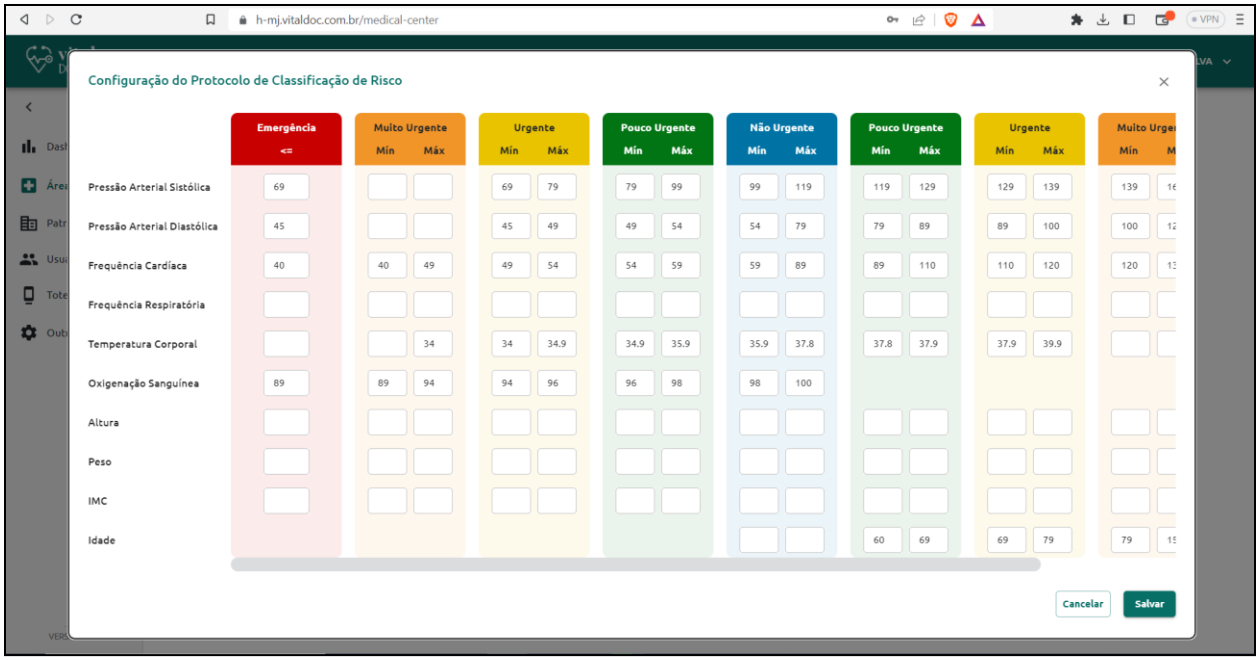

Figura 5: Resultado da Implementação da nova interface de Configuração de Protocolo de Risco

Finalmente, após a conclusão do processo de desenvolvimento, a nova versão da funcionalidade foi implantada no ambiente de produção, sendo disponibilizada para os usuários. A partir desse momento, foram feitos ajustes e melhorias de acordo com o feedback dos usuários, visando garantir a qualidade da funcionalidade e a satisfação dos usuários.

Em resumo, a implementação da solução proposta para a funcionalidade de "Configuração da classificação de risco" foi realizada de forma criteriosa, buscando atender às necessidades e expectativas dos usuários e garantir a qualidade do software.

3.6. Conclusão do Caso Prático

Após a implementação da nova versão da funcionalidade "Configuração da classificação de risco" com base em um levantamento de requisitos mais aprofundado e um processo estruturado de testes e validação, os resultados foram bastante positivos. Os usuários da funcionalidade relataram uma maior facilidade de uso e adequação às suas necessidades reais e passaram a utilizar a funcionalidade nas atividades diárias do ambulatório.

Com base nos problemas identificados no processo de desenvolvimento da primeira versão da funcionalidade, os gestores do setor de Pesquisa e Desenvolvimento decidiram designar uma pessoa dedicada exclusivamente para as atividades de levantamento de requisitos. Isso permitirá um alinhamento mais preciso entre outras funcionalidades desenvolvidas e as

necessidades dos usuários, evitando problemas como os encontrados na primeira versão do software.

Além disso, o processo estruturado de testes e validação da nova versão da funcionalidade contribuiu significativamente para a identificação e correção de problemas antes da implantação do software. Isso aumentou a qualidade do produto entregue aos usuários e reduziu a ocorrência de erros após a implantação.

Em resumo, a experiência de desenvolvimento da funcionalidade "Configuração da classificação de risco" destacou a importância do levantamento de requisitos adequado e da implementação de processos estruturados de teste e validação. A alocação de uma pessoa dedicada exclusivamente para as atividades de levantamento de requisitos mostrou-se uma solução efetiva para evitar problemas de alinhamento entre as funcionalidades desenvolvidas e as necessidades dos usuários finais.

CONCLUSÃO

A partir deste trabalho, foi possível entender a importância do levantamento de requisitos na Pesquisa e Desenvolvimento de Produtos de Software. A falta de uma análise adequada dos requisitos pode gerar diversos problemas, como a insatisfação do cliente, atrasos no projeto e aumento de custos.

Foi apresentado um caso prático que demonstrou como um levantamento de requisitos mais aprofundado e a utilização de técnicas de prototipação podem levar a uma solução mais adequada e satisfatória para os usuários finais.

Além disso, ficou evidente a necessidade de ter uma pessoa dedicada exclusivamente para as atividades de levantamento de requisitos no setor de Pesquisa e Desenvolvimento. Isso pode trazer maior eficiência e qualidade ao processo de desenvolvimento de software.

Portanto, conclui-se que o levantamento de requisitos é uma etapa crucial no desenvolvimento de software e deve ser tratado com a devida importância e atenção pelos profissionais envolvidos no processo. A utilização de técnicas e ferramentas adequadas levará a soluções mais eficientes e satisfatórias para todos os envolvidos no projeto.

REFERÊNCIAS

BARBOSA, S. D. J., & Silva, B. S. (2010). **Interação Humano-Computador**. Elsevier. Disponível em:<https://docplayer.com.br/63299367-Interacao-humano-computador.html>. Acesso em: 10 abr. 2023.

BEATH, C. M., Becerra-Fernandez, I., & Ross, J. W. (2013). **Using the IT Capability Maturity Framework to Drive Business Transformation**. MIS Quarterly Executive, 12(1), 1-19.

KOTONYA, G., & Sommerville, I. (1998). **Requirements engineering:** processes and techniques. Wiley. Disponível em:<https://docplayer.net/216684959-Requirements-engineering-processes-and-techniques-kotonya-pdf.html>. Acesso em: 10 abr. 2023.

LEITE, J. C. S., MEIRA, S. R. L., & FERREIRA, E. C. (2017). **Engenharia de Software:** fundamentos, métodos e padrões. 2ª ed. Elsevier.

MENDONÇA, Ricardo Augusto Ribeiro. **Levantamento de requisitos no desenvolvimento ágil de software.** Goiânia – GO. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/35529111/Levantamento_de_requisitos_no_desenvolvimento_agil_de_software-libre.pdf?1415771233=&response-content-disposition=inline%3B+filename%3DLevantamento_de_requisitos_no_desenvolvi.pdf&Expires=1680368976&Signature=E223wgsnVvS7X3caTqTduB7WiAPPnidgImmvg4Az3Yw2gFXAFSBqKCPTaQO7UlnSuQ1RgqGWfWvEHF5t6L0~WJMaAS8kRwlC6ii57sgrDf6g1BFH2erixbNjS8NjbM0zISbxfhlHbWDWXL8WY0UdNSz7g8MP8is-9w1l-5b8VI3~SJ3pvv6UVhogti4DqICsB~OiSkUr52r~FiuDpo19xsFjf0JCQpuH9xN4hPc-VnVFbDt62-je2A0Za808TRkpGmH8u5g0AhzK6F8dSSpXICAlEbiJRkYmvIA0JEo4T7qjXuPsZeBZHAVp55eZRDNBq6dhtzSG0YElOWseGNb8-A__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. Acesso em: 01 abr. 2023.

PFLEEGER, S. L., & ATLEE, J. M. (2010). **Engenharia de software: teoria e prática**. Pearson Education Brasil. Disponível em: <https://www.bvirtual.com.br/NossoAcervo/Publicacao/476> Acesso em: 28 de Maio de 2023.

POHL, K., RUPP, C., & GERBER, A. (2011). **Requirements Engineering Fundamentals:** A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant. Rocky Nook. Disponível em:<https://www.oreilly.com/pub/pr/2778>. Acesso em: 10 abr. 2023.

PRESSMAN, R. S. (2016). **Engenharia de Software**: uma abordagem profissional. 8ª ed. McGraw-Hill Education. Disponível em: <https://engenhariasoftwareisutic.files.wordpress.com/2016/04/engenharia-software-pressman.pdf>. Acesso em: 05 abr. 2023.

ROBERTSON, S., & ROBERTSON, J. (2012). **Mastering the Requirements Process:** Getting Requirements Right. 3ª ed. Addison-Wesley Professional. Disponível

em:<https://www.oreilly.com/library/view/mastering-the-requirements/9780132942850/>. Acesso em: 16 abr. 2023.

SOMMERVILLE, I. (2011). **Engenharia de Software**. 9ª ed. Pearson. Disponível em:<https://www.academia.edu/42787809/Ian_Sommerville_Engenharia_de_Software_9_ed>. Acesso em: 16 abr. 2023.

SOMMERVILLE, I. (2018). **Engenharia de Software**. 10ª ed. Pearson Education. Disponível em: < https://www.academia.edu/50882590/Sommerville_Software_Engineering_10ed>. Acesso em: 05 abr. 2023.

WIEGERS, K., & BEATTY, J. (2013). **Software Requirements**. 3ª ed. Microsoft Press. Disponível em:<https://ptgmedia.pearsoncmg.com/images/9780735679665/samplepages/9780735679665.pdf>. Acesso em: 16 abr. 2023.